

An Efficient Algorithm to Calculate BICM Capacity

Georg Böcherer*, Fabian Altenbach[§], Alex Alvarado[‡], Steven Corroy[§], and Rudolf Mathar[§]

*Institute for Communications Engineering, Technische Universität München, Germany

Email: georg.boecherer@tum.de

[§]Institute for Theoretical Information Technology, RWTH Aachen University, Germany

Email: {altenbach, corroy, mathar}@ti.rwth-aachen.de

[‡]Signal Processing and Communications Laboratory, University of Cambridge, UK

Email: alex.alvarado@ieee.org

Abstract—Bit-interleaved coded modulation (BICM) is a practical approach for reliable communication over the AWGN channel in the bandwidth limited regime. For a signal point constellation with 2^m points, BICM labels the signal points with bit strings of length m and then treats these m bits separately both at the transmitter and the receiver. BICM capacity is defined as the maximum of a certain achievable rate. Maximization has to be done over the probability mass functions (pmf) of the bits. This is a non-convex optimization problem. So far, the optimal bit pmfs were determined via exhaustive search, which is of exponential complexity in m . In this work, an algorithm called *bit-alternating convex concave method* (BACM) is developed. This algorithm calculates BICM capacity with a complexity that scales approximately as m^3 . The algorithm iteratively applies convex optimization techniques. BACM is used to calculate BICM capacity of 4, 8, 16, 32, and 64-PAM in AWGN. For PAM constellations with more than 8 points, the presented values are the first results known in the literature.

I. INTRODUCTION

Bit-interleaved coded modulation (BICM) [1]–[3] is a *de facto* standard for wireless communications, and it is used in e.g., HSPA, IEEE 802.11a/g/n, and the latest DVB standards (DVB-T2/S2/C2).

In BICM, signal points from a finite constellation are labeled with bit strings. E.g., for 16-PAM, the signal points are labeled with $\log_2 16 = 4$ bits each. The bits in the labels are then treated independently both at the transmitter and the receiver. According to [4], to determine BICM capacity, a certain achievable rate has to be maximized over the bit probability mass functions (pmf). We will make this statement precise later in this work. This maximization is a non-convex optimization problem [5, Fig. 1]. So far, BICM capacity has been calculated using exhaustive search only. For the AWGN channel, results are presented for 8-PAM in [6, Fig. 3] and [5, Fig. 1] and for 16-QAM in [4, Fig. 2]. The complexity of exhaustive search is exponential in the number of bits in the labels, and calculating BICM capacity becomes an intractable problem for large constellations. This motivates the present work.

This work was supported by the German Ministry of Education and Research in the framework of an Alexander von Humboldt Professorship, by the UMIC Research Center, RWTH Aachen University, by The British Academy and The Royal Society (via the Newton International Fellowship scheme), UK, and by the European Community's Seventh's Framework Programme (FP7/2007-2013) under grant agreement No. 271986.

Our approach is as follows. We start by considering a discrete memoryless channel (DMC) operated by a BICM transceiver. To calculate BICM capacity, we develop a new algorithm called *bit alternating convex-concave method* (BACM), which combines two optimization techniques: first, maximization is done sequentially over one bit pmf at a time, and second, the maximization over one bit pmf is done using the convex-concave procedure [7]. We then show how an average power constraint can be taken into account by BACM. This allows us to use BACM to calculate BICM capacity of PAM constellations in AWGN. We provide numerical results for 4 and 8-PAM and, for the first time in the literature, for 16, 32, and 64-PAM. The results show that BICM capacity is close to AWGN capacity and significantly larger than what can be achieved by operating BICM with uniform bit pmfs. Finally, we argue that the complexity of BACM scales approximately as m^3 and logarithmically in the precision with which the optimal bit pmfs are calculated. An implementation of BACM in Matlab is available on our website [8].

II. SYSTEM MODEL AND PROBLEM STATEMENT

Consider a DMC with 2^m input symbols $\mathcal{X} = \{1, \dots, 2^m\}$ and n output symbols $\mathcal{Y} = \{1, \dots, n\}$. The channel is specified by a matrix of transition probabilities $\mathbf{H} \in \mathbf{R}^{n \times 2^m}$, where \mathbf{R} denotes the set of real numbers. The input of the channel is the random variable X , which takes values in \mathcal{X} according to the pmf \mathbf{p} . The channel output is the random variable Y , which takes values in \mathcal{Y} according to the pmf $\mathbf{r} = \mathbf{H}\mathbf{p}$.

A. DMC Capacity

We denote the mutual information between X and Y either by $\mathbb{I}(X; Y)$ or by $\mathbb{I}(\mathbf{p})$. The DMC capacity is [9, Eq. (7.1)]

$$C = \max_{\mathbf{p}} \mathbb{I}(\mathbf{p}). \quad (1)$$

The maximization is a convex optimization problem [10, Prob. 4.57] and it can be solved by the Blahut-Arimoto algorithm [11], [12] or by a software package such as CVX [13].

B. BICM Capacity

In BICM, the input symbols are represented by their m -bit binary expansion, i.e.,

$$\begin{aligned} 1 &\leftrightarrow \overbrace{0 \cdots 00}^{m \text{ bits}} \\ 2 &\leftrightarrow 0 \cdots 01 \\ &\vdots \\ 2^m &\leftrightarrow 1 \cdots 11. \end{aligned} \quad (2)$$

Each bit position of the channel input is treated independently both at the transmitter and the receiver, see [3], [4] for details. This leads to the following constraint at the transmitter:

- [4, Eq. (8)]: The bits B_i in positions i of the channel input are stochastically independent, i.e., the channel input pmf \mathbf{p} is given by

$$\mathbf{p} = \mathbf{p}^1 \otimes \cdots \otimes \mathbf{p}^m \quad (3)$$

where \mathbf{p}^i is the pmf of B_i and where \otimes denotes the Kronecker product, see [14, Def. 4.2.1].

According to [15, Theorem 1], the following sum of mutual informations is an achievable rate for a BICM transceiver:

$$\mathbb{I}^{\text{bicm}}(\mathbf{p}^1, \dots, \mathbf{p}^m) := \sum_{i=1}^m \mathbb{I}(B_i; Y). \quad (4)$$

Following [4, Eq. (19)], the “BICM capacity” C^{bicm} is now given by

$$C^{\text{bicm}} = \max_{\mathbf{p}^1, \dots, \mathbf{p}^m} \mathbb{I}^{\text{bicm}}(\mathbf{p}^1, \dots, \mathbf{p}^m). \quad (5)$$

Unfortunately, the maximization is a non-convex problem. This will become clear in Sec. III.

C. Problem Statement

So far, BICM capacity has been calculated in literature via exhaustive search [4]–[6]. To determine the optimal bit pmfs with a precision of $\pm d$, \mathbb{I}^{bicm} has to be evaluated $(\frac{1}{d})^m$ times, so the complexity of this approach increases exponentially in the number of bit positions m and polynomially in the precision d . The objective of this work is to develop an algorithm that efficiently (compared to exhaustive search) calculates BICM capacity.

III. PRELIMINARY: \mathbb{I}^{bicm} AS A FUNCTION OF \mathbf{p}^i

The goal of this section is to characterize the objective \mathbb{I}^{bicm} as a function of one bit pmf \mathbf{p}^i . By this characterization, it will become clear that \mathbb{I}^{bicm} is a non-convex function, and furthermore, we will see how we can maximize over \mathbf{p}^i . To this end, we pick an arbitrary bit position i and assume that for each $j \neq i$, B_j is distributed according to a fixed pmf and that B_i is distributed according to a pmf that we interpret as a variable. To emphasize this distinction, we denote the pmfs

for $j \neq i$ by $\hat{\mathbf{p}}^j$ and the pmf of B_i by \mathbf{p}^i . The function \mathbb{I}^{bicm} can now be written as

$$\mathbb{I}^{\text{bicm}}(\mathbf{p}^1, \dots, \mathbf{p}^m) = \sum_{j=1}^m [\mathbb{H}(Y) - \mathbb{H}(Y|B_j)] \quad (6)$$

$$= m \mathbb{H}(Y) - \mathbb{H}(Y|B_i) - \sum_{j \neq i} \mathbb{H}(Y|B_j). \quad (7)$$

We see that there are three kinds of terms that we need to express as functions of \mathbf{p}^i : the output entropy $\mathbb{H}(Y)$, the conditional entropy $\mathbb{H}(Y|B_i)$, and the conditional entropy $\mathbb{H}(Y|B_j)$ for $j \neq i$.

A. Output entropy as a function of \mathbf{p}^i

Define

$$\mathbf{q}_0^i := \hat{\mathbf{p}}^1 \otimes \cdots \otimes \hat{\mathbf{p}}^{i-1} \otimes (1 \ 0)^T \otimes \hat{\mathbf{p}}^{i+1} \otimes \cdots \otimes \hat{\mathbf{p}}^m \quad (8)$$

$$\mathbf{q}_1^i := \hat{\mathbf{p}}^1 \otimes \cdots \otimes \hat{\mathbf{p}}^{i-1} \otimes (0 \ 1)^T \otimes \hat{\mathbf{p}}^{i+1} \otimes \cdots \otimes \hat{\mathbf{p}}^m. \quad (9)$$

The channel seen by the i th bit is now given by

$$\mathbf{H}^i = \mathbf{H} \begin{pmatrix} \mathbf{q}_0^i & \mathbf{q}_1^i \end{pmatrix} \in \mathbf{R}^{n \times 2}. \quad (10)$$

The output pmf can now be written as

$$\mathbf{r} = \mathbf{H}\mathbf{p} = \mathbf{H}^i \mathbf{p}^i. \quad (11)$$

Thus, the output entropy as a function of \mathbf{p}^i is given by

$$\mathbb{H}(Y) = - \sum_{k=1}^n r_k \log r_k \quad (12)$$

$$= - \sum_{k=1}^n (\mathbf{r})_k \log (\mathbf{r})_k \quad (13)$$

$$= - \sum_{k=1}^n (\mathbf{H}^i \mathbf{p}^i)_k \log (\mathbf{H}^i \mathbf{p}^i)_k \quad (14)$$

where $(\mathbf{x})_k$ denotes the k th entry of the vector \mathbf{x} . Since $-\mathbf{x} \log \mathbf{x}$ is concave in \mathbf{x} , we conclude that the output entropy is concave in \mathbf{p}^i .

B. Conditional entropy $\mathbb{H}(Y|B_i)$ as a function of \mathbf{p}^i

The output entropy conditioned on the i th bit can be written as

$$\mathbb{H}(Y|B_i) = - \sum_{b=0}^1 p_b^i \sum_{k=1}^n (\mathbf{H}^i)_{kb} \log (\mathbf{H}^i)_{kb} \quad (15)$$

where we index the rows of \mathbf{H}^i by $1, \dots, k$ and the columns by the binary values $0, 1$, e.g., $(\mathbf{H}^i)_{10}$ is the entry of \mathbf{H}^i in the first row and first column. We conclude from (15) that $\mathbb{H}(Y|B_i)$ [and thereby $-\mathbb{H}(Y|B_i)$, which contributes to the objective function] is linear in \mathbf{p}^i .

C. Conditional entropy $\mathbb{H}(Y|B_j)$ as a function of \mathbf{p}^i

Define

$$\mathbf{q}_{00}^{ji} := \hat{\mathbf{p}}^1 \otimes \dots \otimes \hat{\mathbf{p}}^{j-1} \otimes \begin{pmatrix} 1 & 0 \end{pmatrix}^T \otimes \hat{\mathbf{p}}^{j+1} \otimes \dots \otimes \hat{\mathbf{p}}^{i-1} \otimes \begin{pmatrix} 1 & 0 \end{pmatrix}^T \otimes \hat{\mathbf{p}}^{i+1} \otimes \dots \otimes \hat{\mathbf{p}}^m \quad (16)$$

$$\mathbf{q}_{01}^{ji} := \hat{\mathbf{p}}^1 \otimes \dots \otimes \hat{\mathbf{p}}^{j-1} \otimes \begin{pmatrix} 1 & 0 \end{pmatrix}^T \otimes \hat{\mathbf{p}}^{j+1} \otimes \dots \otimes \hat{\mathbf{p}}^{i-1} \otimes \begin{pmatrix} 0 & 1 \end{pmatrix}^T \otimes \hat{\mathbf{p}}^{i+1} \otimes \dots \otimes \hat{\mathbf{p}}^m \quad (17)$$

$$\mathbf{q}_{10}^{ji} := \hat{\mathbf{p}}^1 \otimes \dots \otimes \hat{\mathbf{p}}^{j-1} \otimes \begin{pmatrix} 0 & 1 \end{pmatrix}^T \otimes \hat{\mathbf{p}}^{j+1} \otimes \dots \otimes \hat{\mathbf{p}}^{i-1} \otimes \begin{pmatrix} 1 & 0 \end{pmatrix}^T \otimes \hat{\mathbf{p}}^{i+1} \otimes \dots \otimes \hat{\mathbf{p}}^m \quad (18)$$

$$\mathbf{q}_{11}^{ji} := \hat{\mathbf{p}}^1 \otimes \dots \otimes \hat{\mathbf{p}}^{j-1} \otimes \begin{pmatrix} 0 & 1 \end{pmatrix}^T \otimes \hat{\mathbf{p}}^{j+1} \otimes \dots \otimes \hat{\mathbf{p}}^{i-1} \otimes \begin{pmatrix} 0 & 1 \end{pmatrix}^T \otimes \hat{\mathbf{p}}^{i+1} \otimes \dots \otimes \hat{\mathbf{p}}^m. \quad (19)$$

Now, the channel seen by the j th and the i th bit is given by

$$\mathbf{H}^{ji} = \mathbf{H} \begin{pmatrix} \mathbf{q}_{00}^{ji} & \mathbf{q}_{01}^{ji} & \mathbf{q}_{10}^{ji} & \mathbf{q}_{11}^{ji} \end{pmatrix} \in \mathbb{R}^{n \times 4}. \quad (20)$$

The channel seen by the j th bit can be written as

$$\mathbf{H}^j = \mathbf{H}^{ji} \begin{pmatrix} \mathbf{p}^i & \mathbf{0} \\ \mathbf{0} & \mathbf{p}^i \end{pmatrix}. \quad (21)$$

Thus, the output entropy conditioned on the j th is

$$\mathbb{H}(Y|B_j) = - \sum_{b=0}^1 p_b^j \sum_{k=1}^n (\mathbf{H}^j)_{kb} \log (\mathbf{H}^j)_{kb} \quad (22)$$

$$= - \sum_{b=0}^1 p_b^j \sum_{k=1}^n \left[\mathbf{H}^{ji} \begin{pmatrix} \mathbf{p}^i & \mathbf{0} \\ \mathbf{0} & \mathbf{p}^i \end{pmatrix} \right]_{kb} \log \left[\mathbf{H}^{ji} \begin{pmatrix} \mathbf{p}^i & \mathbf{0} \\ \mathbf{0} & \mathbf{p}^i \end{pmatrix} \right]_{kb}. \quad (23)$$

Since $-x \log x$ is concave in x , we conclude that $\mathbb{H}(Y|B_j)$ is concave in \mathbf{p}^i . As a consequence, the term $-\mathbb{H}(Y|B_j)$, which contributes to the objective function, is convex in \mathbf{p}^i .

D. Summary

The objective function as a function of \mathbf{p}^i can be characterized as follows:

$$\begin{aligned} \mathbb{I}^{\text{bicm}}(\hat{\mathbf{p}}^1, \dots, \hat{\mathbf{p}}^{i-1}, \mathbf{p}^i, \hat{\mathbf{p}}^{i+1}, \dots, \hat{\mathbf{p}}^m) \\ = \underbrace{m \mathbb{H}(Y)}_{\text{concave in } \mathbf{p}^i} - \underbrace{\mathbb{H}(Y|B_i)}_{\text{linear in } \mathbf{p}^i} + \sum_{j \neq i} \underbrace{[-\mathbb{H}(Y|B_j)]}_{\text{convex in } \mathbf{p}^i}. \end{aligned} \quad (24)$$

As a sum of convex and concave terms, \mathbb{I}^{bicm} is a non-convex function. However, as we detail in the next section, the convex-concave procedure [7] can be applied to maximize \mathbb{I}^{bicm} over \mathbf{p}^i .

IV. BACM ALGORITHM

The objective \mathbb{I}^{bicm} is a non-convex function of the pmfs $\mathbf{p}^1, \dots, \mathbf{p}^m$ with potentially more than one local maximum. Thus, finding an efficient algorithm that provably finds the global maximum is difficult. Therefore, we resort to the simpler problem of finding a local maximum. With a good starting point, the global maximum is nevertheless found by such an approach. To find local maxima, efficient methods are available. For the problem at hand, we choose the combination of two methods.

Algorithm 1.(BACM)

```

 $\hat{\mathbf{p}}^1, \dots, \hat{\mathbf{p}}^m \leftarrow$  starting point
repeat bit alternation, outer loop
  for  $i = 1, \dots, m$  bit alternation, inner loop
    maximize  $\mathbb{I}^{\text{bicm}}$  over  $\mathbf{p}^i$  see Alg. 2
    update  $\hat{\mathbf{p}}^i$  with the maximizing  $\mathbf{p}^i$ 
  end for
until convergence

```

Algorithm 2.(convex-concave procedure)

```

calculate  $\mathbf{H}^i$  and  $\mathbf{H}^{ji}$ ,  $j \neq i$ 
 $\mathbf{p}^i \leftarrow \hat{\mathbf{p}}^i$ 
repeat
  1.  $\hat{\mathbf{p}}^i \leftarrow \mathbf{p}^i$ 
  2.  $\mathbf{p}^i \leftarrow \underset{\mathbf{p}^i}{\operatorname{argmax}} f^i(\mathbf{p}^i, \hat{\mathbf{p}}^i)$  see Subsec. IV-B
until convergence

```

- We maximize over one bit pmf \mathbf{p}^i at a time and then cycle through the $i = 1, \dots, m$ until convergence. This approach goes under the name *alternating maximization*.
- To maximize over one bit pmf \mathbf{p}^i , we iteratively approximate \mathbb{I}^{bicm} by a lower bound that is concave in \mathbf{p}^i and maximize this concave lower bound. After convergence, the maximum of the concave lower bound is also a local maximum of \mathbb{I}^{bicm} as a function of \mathbf{p}^i . This technique is known as the convex-concave procedure [7].

We call this approach the *bit-alternating convex-concave method* (BACM). The alternating maximization over the bit pmfs is displayed in Alg. 1. The maximization over one bit pmf is detailed next.

A. Concave Lower Bound

As the objective is the sum of concave and convex functions, it cannot be maximized directly. However, the *convex-concave procedure* as defined in [16, slide 26] can be applied. Define the function $h^j(\mathbf{p}^i)$ as the negative of the right-hand side of (23). This function is convex in \mathbf{p}^i . The convex-concave procedure is an iterative method and works as follows. Denote by $\hat{\mathbf{p}}^i$ the result for \mathbf{p}^i in the previous step. Then, in the current step, approximate $h^j(\mathbf{p}^i)$ by its first order Taylor expansion in $\hat{\mathbf{p}}^i$, i.e., by

$$\hat{h}^j(\mathbf{p}^i, \hat{\mathbf{p}}^i) := h^j(\hat{\mathbf{p}}^i) + \nabla h^j(\hat{\mathbf{p}}^i)^T (\mathbf{p}^i - \hat{\mathbf{p}}^i). \quad (25)$$

Note that since $h^j(\mathbf{p}^i)$ is convex in \mathbf{p}^i and the approximation $\hat{h}^j(\mathbf{p}^i, \hat{\mathbf{p}}^i)$ is linear in \mathbf{p}^i , the approximation $\hat{h}^j(\mathbf{p}^i, \hat{\mathbf{p}}^i)$ lower bounds $h^j(\mathbf{p}^i)$ for any value of \mathbf{p}^i . By a calculation similar to [17, (7.61)–(7.63)] it can be shown that \hat{h}^j is given by

$$\begin{aligned} \hat{h}^j(\mathbf{p}^i, \hat{\mathbf{p}}^i) = \\ \sum_{b=0}^1 p_b^j \sum_{k=1}^n \left[\mathbf{H}^{ji} \begin{pmatrix} \mathbf{p}^i & \mathbf{0} \\ \mathbf{0} & \mathbf{p}^i \end{pmatrix} \right]_{kb} \log \left[\mathbf{H}^{ji} \begin{pmatrix} \hat{\mathbf{p}}^i & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{p}}^i \end{pmatrix} \right]_{kb}. \end{aligned} \quad (26)$$

Putting all together, we have a concave lower bound of \mathbb{I}^{bicm} as a function of \mathbf{p}^i given by

$$f^i(\mathbf{p}^i, \hat{\mathbf{p}}^i) := m \mathbb{H}(Y) - \mathbb{H}(Y|B_i) + \sum_{j \neq i} \hat{h}^j(\mathbf{p}^i, \hat{\mathbf{p}}^i) \quad (27)$$

$$\begin{aligned} &= -m \sum_{k=1}^n (\mathbf{H}^i \mathbf{p}^i)_k \log(\mathbf{H}^i \mathbf{p}^i)_k + \sum_{b=0}^1 p_b^i \sum_{k=1}^n (\mathbf{H}^i)_{kb} \log(\mathbf{H}^i)_{kb} \\ &+ \sum_{j \neq i} \sum_{b=0}^1 p_b^j \sum_{k=1}^n \left[\mathbf{H}^{ji} \begin{pmatrix} \mathbf{p}^i & \mathbf{0} \\ \mathbf{0} & \mathbf{p}^i \end{pmatrix} \right]_{kb} \log \left[\mathbf{H}^{ji} \begin{pmatrix} \hat{\mathbf{p}}^i & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{p}}^i \end{pmatrix} \right]_{kb} \end{aligned} \quad (28)$$

Since f^i is a concave function of \mathbf{p}^i , it can be maximized efficiently over \mathbf{p}^i , as we will explain in detail in the next subsection. We iteratively update $\hat{\mathbf{p}}^i$ with the value of \mathbf{p}^i that maximizes $f^i(\mathbf{p}^i, \hat{\mathbf{p}}^i)$. Algorithm 2 illustrates this procedure. After convergence, the pmf \mathbf{p}^i locally maximizes \mathbb{I}^{bicm} over \mathbf{p}^i given the fixed pmfs $\hat{\mathbf{p}}^j$ for $j \neq i$.

B. Solving the Inner Optimization Problem

We need to solve the optimization problem

$$\underset{\text{pmf } \mathbf{p}^i}{\text{maximize}} f^i(\mathbf{p}^i, \hat{\mathbf{p}}^i). \quad (29)$$

Any pmf \mathbf{p}^i can for some $p_0^i \in [0, 1]$ be written as $\mathbf{p}^i = \begin{bmatrix} p_0^i \\ 1-p_0^i \end{bmatrix}$. We define

$$f_0^i(p_0^i, \hat{\mathbf{p}}^i) = f^i\left(\begin{bmatrix} p_0^i \\ 1-p_0^i \end{bmatrix}, \hat{\mathbf{p}}^i\right). \quad (30)$$

We can now formulate our optimization problem as

$$\underset{p_0^i \in [0, 1]}{\text{maximize}} f_0^i(p_0^i, \hat{\mathbf{p}}^i). \quad (31)$$

Note that the problems (29) and (31) are equivalent and furthermore, by [10, Sec. 3.2.2], f_0^i is a concave function of p_0^i . Thus, our problem reduces to finding the maximum of a concave function with a scalar argument. This can be done as follows.

The first derivative of $H(Y)$, $H(Y|B_i)$, and $\hat{h}^j(\begin{bmatrix} p_0^i \\ 1-p_0^i \end{bmatrix}, \hat{\mathbf{p}}^i)$, $j \neq i$ with respect to p_0^i are respectively given by

$$\begin{aligned} \frac{\partial H(Y)}{\partial p_0^i} &= - \sum_{k=1}^n \left\{ \left[\mathbf{H}^i \begin{pmatrix} 1 & -1 \end{pmatrix}^T \right]_k \log(\mathbf{H}^i \begin{bmatrix} p_0^i \\ 1-p_0^i \end{bmatrix})_k \right. \\ &\quad \left. + \left[\mathbf{H}^i \begin{pmatrix} 1 & -1 \end{pmatrix}^T \right]_k \right\} \end{aligned} \quad (32)$$

$$\frac{\partial H(Y|B_i)}{\partial p_0^i} = \sum_{k=1}^n [(\mathbf{H}^i)_{k0} \log(\mathbf{H}^i)_{k0} - (\mathbf{H}^i)_{k1} \log(\mathbf{H}^i)_{k1}] \quad (33)$$

$$\begin{aligned} \frac{\partial \hat{h}^j(\begin{bmatrix} p_0^i \\ 1-p_0^i \end{bmatrix}, \hat{\mathbf{p}}^i)}{\partial p_0^i} &= \sum_{b=0}^1 p_b^j \sum_{k=1}^n [(\mathbf{H}^{ji})_{kb0} - (\mathbf{H}^{ji})_{kb1}] \\ &\quad \cdot \log(\mathbf{H}^j)_{kb} \end{aligned} \quad (34)$$

where we index the rows of \mathbf{H}^{ji} by $k = 1, \dots, n$ and the columns by the binary expansion $b_j b_i = 00, 01, 10, 11$.

E.g., $(\mathbf{H}^{ji})_{110}$ denotes the entry of \mathbf{H}^{ji} in the 1st row and the 3rd column. For notational convenience, we write

$$df_0^i(p_0^i, \hat{\mathbf{p}}^i) := \frac{\partial f_0^i(p_0^i, \hat{\mathbf{p}}^i)}{\partial p_0^i}. \quad (35)$$

Putting the expressions above together according to (27), we get the first derivative of f_0^i . Since f_0^i is concave, df_0^i is monotonically decreasing in p_0^i . Consequently, we can maximize f_0^i over $p_0^i \in [0, 1]$ as follows.

$$\underset{p_0^i}{\text{argmax}} f_0^i(p_0^i, \hat{\mathbf{p}}^i) = \begin{cases} 0 & df_0^i(0^+, \hat{\mathbf{p}}^i) < 0 \\ 1 & df_0^i(1^-, \hat{\mathbf{p}}^i) > 0 \\ p_0^i : df_0^i(p_0^i, \hat{\mathbf{p}}^i) = 0 & \text{otherwise.} \end{cases} \quad (36)$$

In our implementation [8], we use the bisection method to find p_0^i in the third case. See Sec. VII for details.

V. ADDING AN AVERAGE COST CONSTRAINT

We discuss how BACM can be used to calculate BICM capacity when the bit pmfs are subject to an average cost constraint. Suppose we have a cost vector $\mathbf{w} \in \mathbf{R}_{>0}^{2^m}$, where $\mathbf{R}_{>0}$ denotes the set of positive real numbers. Then, the symbol costs seen by the i th bit are given by

$$\mathbf{w}^i = [\mathbf{w}^T(\mathbf{q}_0^i \quad \mathbf{q}_1^i)]^T. \quad (37)$$

The average cost can now be included by adding a weighted version of the average cost $\mathbf{w}^{iT} \mathbf{p}^i$ to f^i , i.e., the inner optimization problem in Alg. 2 now becomes

$$\underset{\mathbf{p}^i}{\text{maximize}} [f^i(\mathbf{p}^i, \hat{\mathbf{p}}^i) - \lambda \mathbf{w}^{iT} \mathbf{p}^i]. \quad (38)$$

This simply adds another linear term and our algorithm works in exactly the same way as before. Denote by \mathbf{p}^{i*} the optimal pmfs found by this modified version of BACM for some λ . Consider the resulting cost

$$E = \mathbf{w}^T \mathbf{p}^* \quad (39)$$

where $\mathbf{p}^* = \mathbf{p}^{1*} \otimes \dots \otimes \mathbf{p}^{m*}$. Then, it can be shown that the bit pmfs $\mathbf{p}^{1*}, \dots, \mathbf{p}^{m*}$ solve the optimization problem

$$\begin{aligned} &\underset{\mathbf{p}^1, \dots, \mathbf{p}^m}{\text{maximize}} \quad \mathbb{I}^{\text{bicm}}(\mathbf{p}^1, \dots, \mathbf{p}^m) \\ &\text{subject to} \quad \mathbf{w}^T(\mathbf{p}^1 \otimes \dots \otimes \mathbf{p}^m) \leq E. \end{aligned} \quad (40)$$

VI. APPLICATION TO PAM IN AWGN

We use BACM to calculate BICM capacity of PAM constellations in AWGN. To calculate the BICM capacity of PAM constellations in AWGN, optimization has to be done over the labeling of the signal points, the scaling of the constellation, and the bit pmfs, see [6, Eq. (40)] for details. Here, we fix the labeling to the binary reflected Gray code [6, Sec. II-B] and optimize over constellation scaling and bit pmfs. To be able to use BACM, we discretize the channel output into 200 equally spaced points. For each scaling, the discretized AWGN channel with $M = 2^m$ constellation points at the input can thus be represented by a DMC specified by a transition matrix $\mathbf{H} \in \mathbf{R}^{200 \times M}$. For this DMC, we

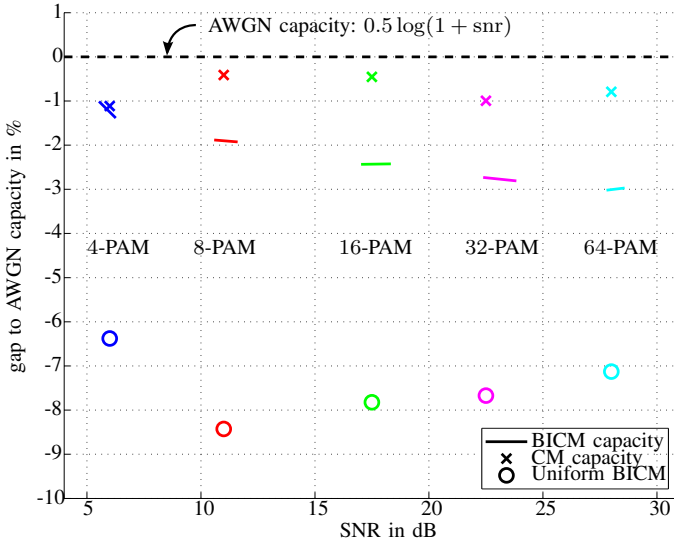


Fig. 1. Results for 4, 8, 16, 32, and 64-PAM in AWGN. In the horizontal direction, SNR is displayed in dB. In the vertical direction, we show the gap in percent to the AWGN capacity $C(\text{snr}) = 0.5 \log(1 + \text{snr})$. E.g., for BICM capacity, the gap is calculated as $100 \cdot (1 - \frac{C(\text{snr})}{C^{\text{bicm}}(\text{snr})})$. For each constellation size and a corresponding target SNR, CM capacity, BICM capacity, and uniform BICM capacity are displayed. For BICM capacity, we display several values since we could adjust the effective SNR only via the weighting factor λ , see Sec. V.

use the method proposed in Sec. V to calculate the BICM capacity. To achieve a target SNR, we iteratively adapt the weighting λ of the average power in (38). We repeat this for different constellation scalings and choose the scaling that yields the largest value for \mathbb{I}^{bicm} . This largest value is the BICM capacity and we denote it by $C^{\text{bicm}}(\text{snr})$. Results for 4, 8, 16, 32, and 64-PAM are displayed in Fig. 1. For comparison, coded modulation (CM) capacity [6, Eq. (28)] of the corresponding constellation and \mathbb{I}^{bicm} for uniform bit pmfs are displayed. The values for CM capacity were obtained via CVX [13]. The BICM capacity significantly outperforms uniform BICM and gets close to CM capacity. We calculated the optimal bit pmfs with a precision of $d = 10^{-5}$.

VII. COMPLEXITY OF BACM

We start by analyzing the complexity of the inner optimization problem. To cover the first two cases in (36), we need to evaluate df_0^i two times. To find the p_0^i in the third case we use the bisection method starting with the upper bound $u = 1$ and the lower bound $\ell = 0$, and we terminate when $u - \ell \leq 2d$. After termination, we assign $p_0^i = \frac{u+\ell}{2}$. Thus, we calculate p_0^i with a precision of $\pm d$. According to [10, p. 146], the number of times we need to evaluate df_0^i until termination is given by

$$\left\lceil \log_2 \frac{u - \ell}{2d} \right\rceil = \left\lceil \log_2 \frac{1 - 0}{2d} \right\rceil = \left\lceil \log_2 \frac{1}{2d} \right\rceil. \quad (41)$$

When evaluating df^i , by (27), we need to evaluate $\partial \hat{h}^j / \partial p_0^i$ for each $j \neq i$, which results in a number of $m - 1$ or roughly m evaluations. Overall, the number of evaluations needed for solving the inner optimization problem once is roughly $m \log_2 \frac{1}{2d}$. The sizes of the matrices involved in (28) are invariant under m , i.e., $\mathbf{H}^{ji} \in \mathbb{R}^{n \times 4}$ and $\mathbf{H}^i \in \mathbb{R}^{n \times 2}$.

Therefore, the number of iterations until convergence in Alg. 2 should be approximately invariant under m and we denote it by a constant K . For our AWGN simulations, this number was around $K = 3$, independent of m . The complexity of maximizing \mathbb{I}^{bicm} over one bit pmf is thus approximately $Km \log_2 \frac{1}{2d}$. This maximization has to be done for $i = 1, \dots, m$, i.e., m times, which adds another factor of m to the complexity. This procedure has to be repeated L times until convergence in the outer loop of Alg. 1. This number depends on m . For the AWGN simulations, we observed for $m = 2, 3, 4, 5, 6$, respectively, the values

$$2.00 \quad 3.27 \quad 3.90 \quad 4.24 \quad 4.31. \quad (42)$$

The average for each m is taken separately over all values that were observed when executing BACM. This value increases slightly with m . To have a rough bound on complexity, we assume that L increases at most linearly with m , which is consistent with the observed data (42). All together, we have a complexity that is approximately of the order $LKm^2 \log_2 \frac{1}{2d} \leq Km^3 \log_2 \frac{1}{2d}$. In summary, BACM scales as m^3 and logarithmically in the precision d .

REFERENCES

- [1] E. Zehavi, "8-PSK trellis codes for a Rayleigh channel," *IEEE Trans. Commun.*, vol. 40, no. 3, pp. 873–884, May 1992.
- [2] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 927–946, May 1998.
- [3] A. Guillén i Fàbregas, A. Martinez, and G. Caire, "Bit-interleaved coded modulation," *Found. Trends Comm. Inf. Theory*, vol. 5, no. 1–2, pp. 1–153, 2008.
- [4] A. Guillén i Fàbregas and A. Martinez, "Bit-interleaved coded modulation with shaping," in *Proc. IEEE Information Theory Workshop (ITW)*, 2010, pp. 1–5.
- [5] A. Alvarado, F. Brännström, and E. Agrell, "High SNR bounds for the BICM capacity," in *Proc. IEEE Information Theory Workshop (ITW)*, 2011, pp. 360–364.
- [6] E. Agrell and A. Alvarado, "Optimal alphabets and binary labelings for BICM at low SNR," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6650–6672, Oct. 2011.
- [7] A. L. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural Computation*, vol. 15, pp. 915–936, 2003.
- [8] "BACM—an algorithm for calculating BICM capacity." [Online]. Available: <http://www.georg-boecherer.de/bacm>
- [9] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, Inc., 2006.
- [10] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [11] R. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. Inf. Theory*, vol. 18, no. 4, pp. 460–473, 1972.
- [12] S. Arimoto, "An algorithm for computing the capacity of arbitrary discrete memoryless channels," *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 14–20, 1972.
- [13] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," <http://cvxr.com/cvx>, Jul. 2010.
- [14] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, 1991.
- [15] A. Martinez, A. Guillén i Fàbregas, G. Caire, and F. Willems, "Bit-interleaved coded modulation revisited: A mismatched decoding perspective," vol. 55, no. 6, pp. 2756–2765, 2009.
- [16] S. Boyd, "Convex optimization II, lecture 14: Sequential convex programming," lecture notes, 2008. [Online]. Available: http://www.stanford.edu/class/ee364b/lectures/seq_slides.pdf
- [17] G. Böcherer, "Capacity-achieving probabilistic shaping for noisy and noiseless channels," Ph.D. dissertation, RWTH Aachen University, 2012. [Online]. Available: <http://www.georg-boecherer.de/capacityAchievingShaping.pdf>